Lecture # 11 memory virtualization.

sys_spawn() {

fork();

}

os fn.

spawn() {

fork();

fork();

:

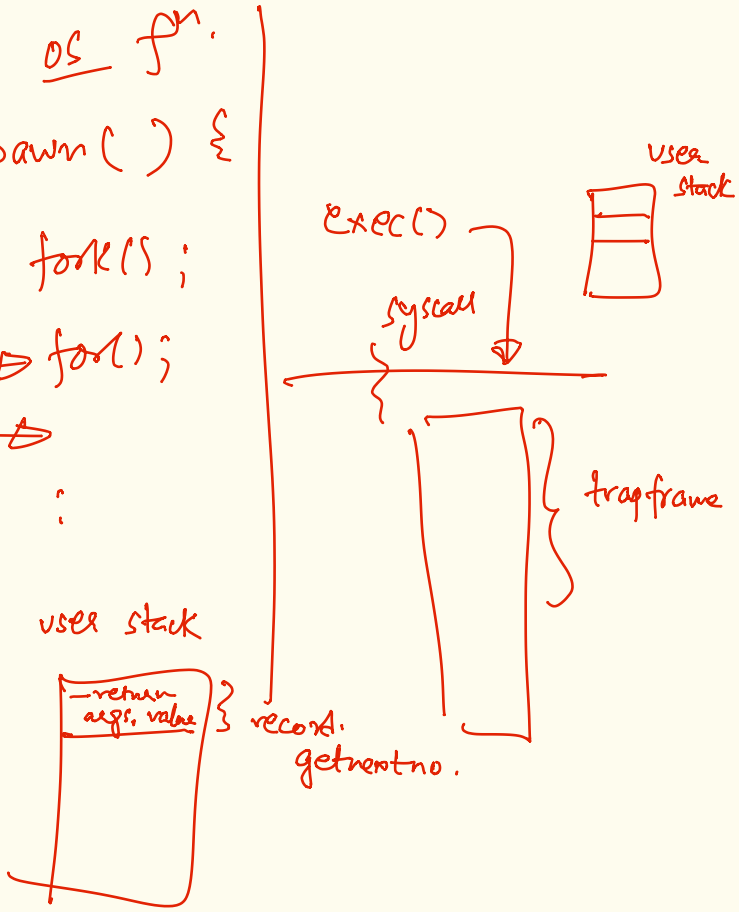
}

main {

getnextnumber();

return 0;

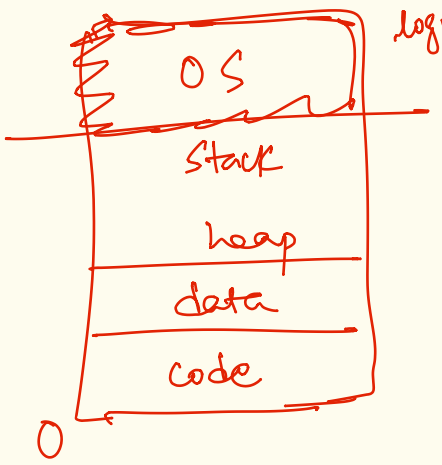
}



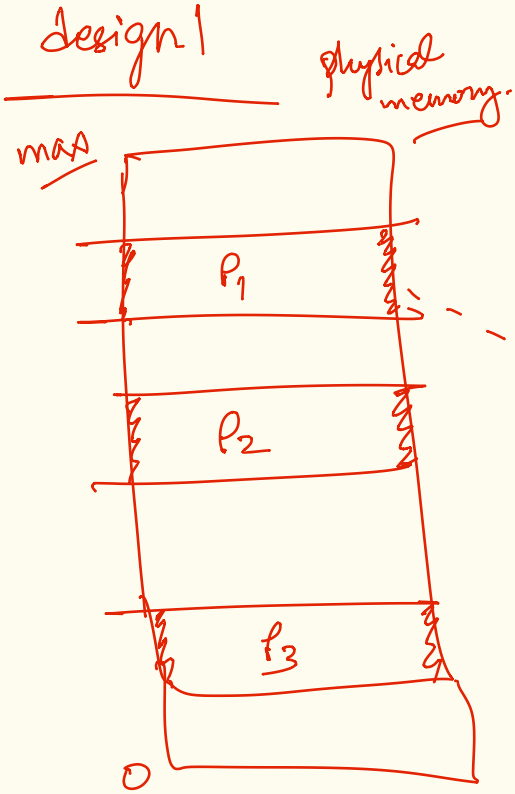
memory virtualization

- isolation
- full addressability
- efficiency

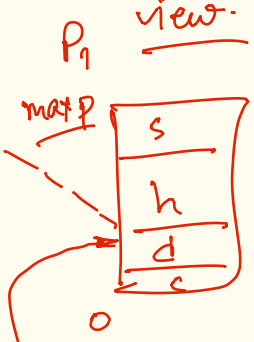
design 0



design 1



process logical view.



max p < mem capacity

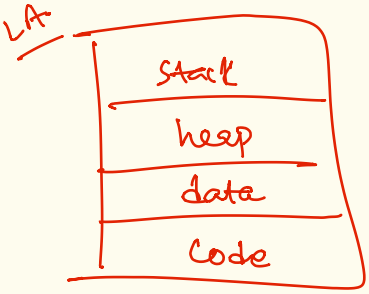
int a = 23; // &a;

② design 2

segmentation.

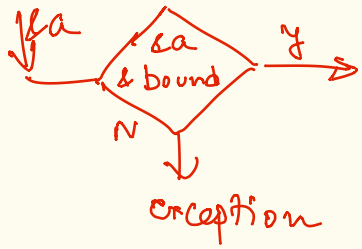
- -ves of design 1
- internal & extend fragmentation

- breaks process address space into segments, each of which are independently mapped in physical memory.



per process base register & bound bound.

int a = 23



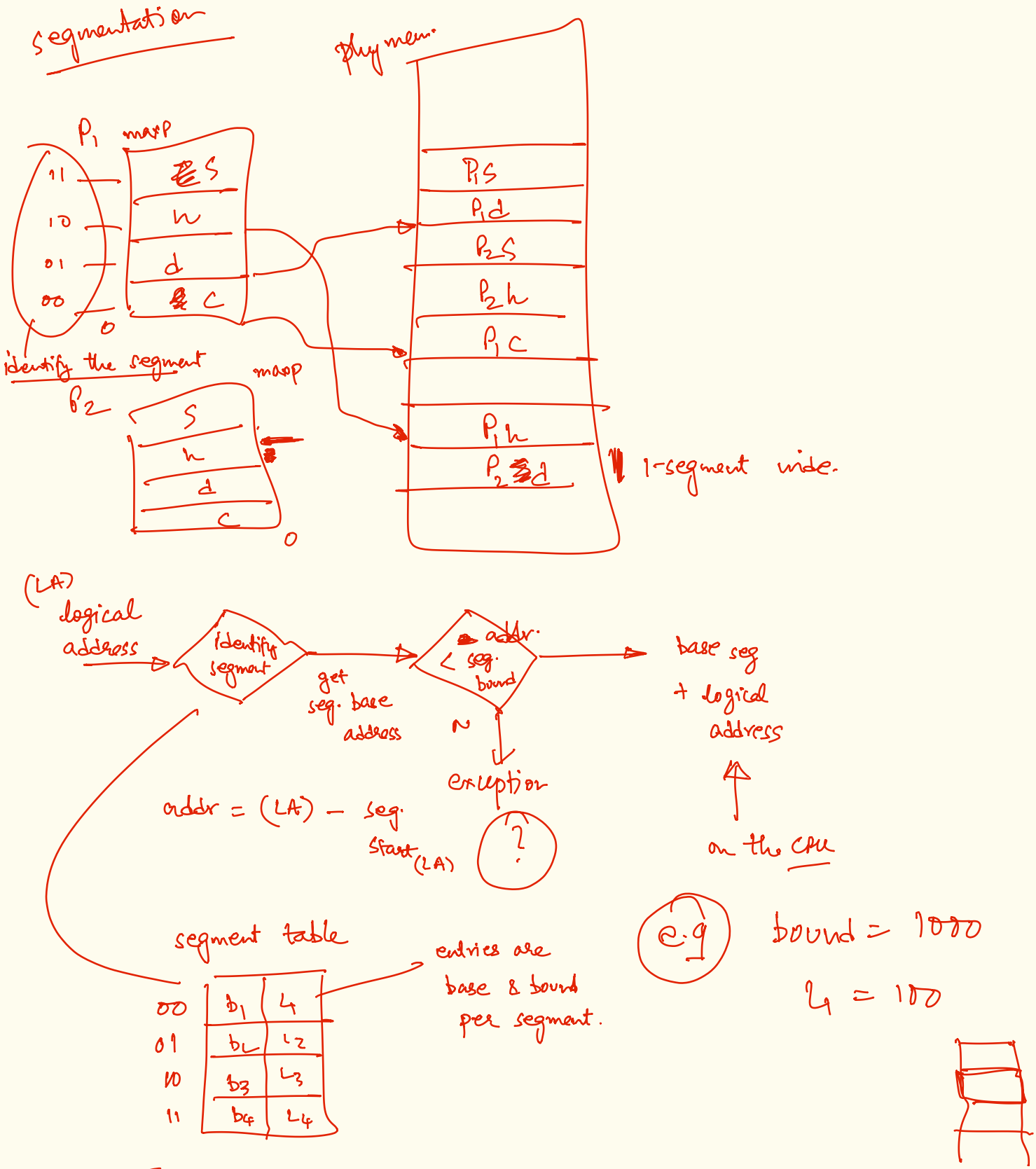
PA to access

- this translation in in hw
- mmu — memory mgmt. unit

*(base + &a) = 23;

this address is used by CPU on the address bus.

- base & bound are per process metadata.
- part of content of process
- stored in PCB.



design 3: segments ~ are still contiguous!

paging

~ restrict unit of memory to 1 page

4KB ⇐ # addresses / # bytes